An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 42 (2001) 1311–1317
www.elsevier.nl/locate/camwa

An Efficient Method for l^∞ Regression

JUN JI AND C. KICEY
Department of Mathematics and Computer Science
Valdosta State University, Valdosta, GA 31698, U.S.A.
⟨junji⟩⟨ckicey⟩@valdosta.edu

(Received March 2000; revised and accepted February 2001)

Abstract—Using a few very basic observations, we take full advantage of the special structure of the l^∞ regression problem to obtain a direct and finite algorithm for the computation of the l^∞ regression line. The complexity of the algorithm and the computational results showing the effectiveness of the algorithm will also be presented. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Algorithm, Complexity, Linear regression, Optimization.

1. INTRODUCTION

Linear regression is a very popular topic not only in mathematics, but also in the natural and social sciences and engineering. However, there are many methods to find a "best-fit" or regression line. To begin, suppose we have a number of points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, perhaps collected from some scientific experiment, and we seek a linear function $f(x) = mx + b$ such that $f(x_k) = y_k$ for all $k = 1, 2, \dots, n$. Of course, due to the very nature of experimentation and data collection, we cannot expect any line to pass through all of the points. Instead, we try to find a line $f(x) = mx + b$ which is as close to the data points as possible. Usually, this is done by making all the *residuals*

$$r_k = |y - f(x_k)|, \qquad k = 1, 2, \dots, n$$

simultaneously small. This leads to the least-norm problem of finding $f(x) = mx + b$ such that the norm of the vector of residuals $\mathbf{r} = (r_1, r_2, \dots, r_n)$ is minimized. If we choose the Euclidean norm, then we have a smooth optimization problem and the regression obtained is often called the *least-squares regression*. In a variety of real applications, least squares is not a good measure of distance and other regression methods with other norms are of interest. For example, Gibson and Pulapaka [1] used l^1 regression while developing a log rotation algorithm for Langdale Forest Products Company™. If the Euclidean norm is replaced by l^1 or l^∞, then the resulting regression is a nonsmooth optimization problem and is more difficult to compute than the least squares regression [2–4]. By introducing a few new variables, the l^1 and l^∞ regression were often converted into LP problems and solved through the simplex method [5]. Along this line, an algorithm for the solution of the l^∞ regression in a general setting was proposed in [3] and a more general approach can be found in [4].

The authors are grateful to anonymous referees for their valuable comments on the earlier version of the paper which substantially improve the presentation of the paper.

0898-1221/01/$ - see front matter © 2001 Elsevier Science Ltd. All rights reserved. Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(01)00242-5

In this paper, we will focus on the geometric structure of the l^∞ regression problem. By taking full advantage of its special structure, together with a few very simple observations, we will propose a direct and finite algorithm, combinatorial in nature, for computing the l^∞ regression line. We will also study the complexity of the algorithm which, together with our computational experience with the algorithm, indicates the efficiency and robustness of the algorithm.

2. TECHNICAL RESULTS

To start, we explore the structure of the problem and collect some basic facts involved in this problem. The key to our approach is the following simple observation.

LEMMA 1. *Suppose that $c_1 \le c_2 \le \cdots \le c_n$ and let*

$$f(x) = \max\{|x - c_1|, |x - c_2|, \dots, |x - c_n|\}.$$

Then the minimum value of $f(x)$ over $(-\infty, +\infty)$ occurs at $x = (c_1 + c_n)/2$ with value $(c_n - c_1)/2$.

PROOF. For every real number x, we have $c_n - c_1 = |(c_n - x) + (x - c_1)| \le |x - c_n| + |x - c_1|$, which implies $f(x) \ge \max\{|x - c_1|, |x - c_n|\} \ge (c_n - c_1)/2$. Also, for all $i = 1, 2, \dots, n$, we have

$$-\frac{(c_n - c_1)}{2} = c_1 - \frac{(c_n + c_1)}{2} \le c_i - \frac{(c_n + c_1)}{2} \le c_n - \frac{(c_n + c_1)}{2} = \frac{(c_n - c_1)}{2}.$$

It now follows that $f((c_1 + c_n)/2) = (c_n - c_1)/2$ and the proof is complete.

We begin with a set of data points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. For simplicity of our presentation, we assume $x_1 < x_2 < \cdots < x_n$ are distinct. The other cases will be discussed later in the text. Finding the l^∞ regression line $y = mx + b$ is equivalent to finding a pair (m^*, b^*) that minimizes

$$R(m, b) = \max\{|mx_1 + b - y_1|, |mx_2 + b - y_2|, \dots, |mx_n + b - y_n|\}. \tag{1}$$

In order to take advantage of Lemma 1, we rewrite $R(m, b)$ as

$$R(m, b) = \max\{|b - (y_1 - x_1 m)|, |b - (y_2 - x_2 m)|, \dots, |b - (y_n - x_n m)|\}. \tag{2}$$

Define

$$\mathcal{F} \equiv \{f_k(m) = y_k - x_k m : k = 1, 2, \dots, n\}.$$

For each $f_k(m) \in \mathcal{F}$, $z = f_k(m)$ defines a straight line, denoted by L_k, in the m-z-plane, and the family of lines $\{L_k : k = 1, 2, \dots, n\}$ divides the m-z-plane into a finite number of polygonal regions. Define

$$\phi_1(m) = \min_k\{f_k(m) : k = 1, \dots, n\} \qquad \text{and} \qquad \phi_2(m) = \max_k\{f_k(m) : k = 1, \dots, n\}.$$

It is easily seen that both $\phi_1(m)$ and $\phi_2(m)$ are continuous piecewise linear functions. Let P and Q be the sets of vertices of the graphs of functions of $z = \phi_1(m)$ and $z = \phi_2(m)$, respectively. Let $M = \{m_1, m_2, \dots, m_t\}$ be the collection of all the distinct m-coordinates of $P \cup Q$. Without loss of generality, we assume that $m_1 < m_2 < \cdots < m_t$. For convenience, we define $m_0 = -\infty$ and $m_{t+1} = +\infty$. We also denote the interval $[m_i, m_{i+1}]$ by I_i for all $1 \le i \le t-1$, $I_0 = (-\infty, m_1]$, and $I_t = [m_t, \infty)$. For the interval I_i, we will say that the lines $L_{I_i^-}$ and $L_{I_i^+}$ are, respectively, the *floor* and *ceiling* of $\mathcal{F}$ on I_i, if

$$f_{I_i^-}(m) \le f_k(m) \le f_{I_i^+}(m), \qquad \text{for all } m \in I_i \text{ and } k = 1, 2, \dots, n.$$

Here and throughout the paper, we use I_i^- and I_i^+ to denote the indices of the floor and the ceiling of $\mathcal{F}$ on I_i. Clearly, the monotonicity of x_i implies that L_1 and L_n are the floor and ceiling of $\mathcal{F}$ on the interval I_0, respectively. Similarly, L_n and L_1 are the floor and ceiling of $\mathcal{F}$ on the interval I_t, respectively. That is, we have

$$I_0^- = 1, \qquad I_0^+ = n, \qquad I_t^- = n, \qquad \text{and} \qquad I_t^+ = 1.$$

Lemma 1 implies that for $m \in I_i$,

$$\min_b R(m,b) = \frac{f_{I_i^+}(m) - f_{I_i^-}(m)}{2} = \frac{\left(x_{I_i^-} - x_{I_i^+}\right)m + \left(y_{I_i^+} - y_{I_i^-}\right)}{2} \equiv \psi_i(m), \tag{3}$$

when

$$b = b_i(m) = \frac{y_{I_i^-} + y_{I_i^+} - m\left(x_{I_i^-} + x_{I_i^+}\right)}{2}. \tag{4}$$

For any m in $(-\infty, +\infty)$, we define $\psi(m) = \min_b R(m,b)$. Clearly, $\psi(m) = \psi_i(m)$ on I_i, $i = 0, 1, \dots, t$. It is easily seen from (3) that $\psi(m)$ is a line segment on I_i, and therefore, $\psi(m)$ is strictly increasing on I_i if $I_i^- > I_i^+$ and strictly decreasing on I_i if $I_i^- < I_i^+$. Consequently, its minimum value over $I_i = [m_i, m_{i+1}]$ must occur at one of the endpoints m_i or m_{i+1}. To be more specific, we have

$$\min_{m \in I_i} \psi(m) = \begin{cases} \psi(m_{i+1}), & \text{if } I_i^- < I_i^+, \\ \psi(m_i), & \text{if } I_i^- > I_i^+. \end{cases} \tag{5}$$

Again, the monotonicity of x_i implies that if L_r intersects L_s $(r > s)$ at $(\bar{m}, \bar{z})$, then

$$\max\{f_r(m), f_s(m)\} = \begin{cases} f_r(m), & \text{if } m \le \bar{m}, \\ f_s(m), & \text{if } m > \bar{m}, \end{cases} \tag{6}$$

$$\min\{f_r(m), f_s(m)\} = \begin{cases} f_s(m), & \text{if } m \le \bar{m}, \\ f_r(m), & \text{if } m > \bar{m}. \end{cases} \tag{7}$$

This simple observation leads us to the following result.

LEMMA 2. *The index of the floors, I_i^-, is an increasing function of i while the index of the ceiling, I_i^+, is a decreasing function of i for $i = 0, 1, \dots, t$.*

From (3), we see that $\psi(m)$ is a piecewise linear function over $(-\infty, +\infty)$. Also, (5) implies that its minimizer must be at one of these m_k, $k = 1, 2, \dots, t$. To this end, we have the following result.

THEOREM 1. *There exists a unique minimizer m_K of $z = \psi(m)$ over $(-\infty, \infty)$ where K satisfies*

$$I_i^- < I_i^+, \qquad \text{for all } 0 \le i \le K-1 \text{ and } I_K^- \ge I_K^+. \tag{8}$$

PROOF. Lemma 2 and (8) imply that

$$I_i^- \ge I_K^- \ge I_K^+ \ge I_i^+, \qquad \text{for all } K \le i \le t,$$

which, together with (5) and the fact that $\psi(m)$ is monotonic on each I_i, further implies that $\psi(m)$ is decreasing on $(-\infty, m_K]$ while increasing on $[m_K, +\infty)$. Thus, $\psi(m)$ is minimized at m_K for the K satisfying (8). The uniqueness of such an m_K follows directly from the strict monotonity of x_i, $i = 1, 2, \dots, n$.

3. THE ALGORITHM FOR l^∞ LINEAR REGRESSION

The algorithm presented in this section will search for the m_K among these m_i, $i = 1, \dots, t$, starting with m_1. One typical step of our algorithm is to get m_{i+1}, I_{i+1}^-, and I_{i+1}^+ from m_i, I_i^- and I_i^+. To describe one typical step, let us define

$$m_{i,j} = \frac{y_i - y_j}{x_i - x_j}, \quad m_i^- = \min_j \left\{ m_{I_i^-,j} : j > I_i^- \right\}, \quad \text{and} \quad m_i^+ = \min_j \left\{ m_{I_i^+,j} : j < I_i^+ \right\}. \tag{9}$$

The $m_{i,j}$ are actually the m-coordinates of the intersections of the straight lines L_i and L_j and m_i^- and m_i^+ are the m-coordinates of the vertices of the graphs of the floor $z = \phi_1(m)$ and the ceiling $z = \phi_2(m)$, respectively. It is easily seen from Lemma 2, (6), and (7) that the smaller of m_i^- and m_i^+ defines m_{i+1}. Also, the incoming index for the floor of $\mathcal{F}$ on I_{i+1} is the largest index among those binding at m_{i+1}, while the incoming index for the ceiling of $\mathcal{F}$ on I_{i+1} is the smallest index among those binding at m_{i+1}. More precisely, we have

$$m_{i+1} = \min \left\{ m_i^-, m_i^+ \right\}, \qquad \text{for } i = 0, 1, \dots, t-1, \tag{10}$$

$$I_{i+1}^- = \begin{cases} I_i^-, & \text{if } m_i^+ < m_i^-, \\ \max \left\{ p : m_i^- = m_{I_i^-,p},\ p > I_i^- \right\}, & \text{if } m_i^+ \geq m_i^-, \end{cases} \tag{11}$$

$$I_{i+1}^+ = \begin{cases} \min \left\{ q : \ m_i^+ = m_{I_i^+,q},\ q < I_i^+ \right\}, & \text{if } m_i^+ \leq m_i^-, \\ I_i^+, & \text{if } m_i^+ > m_i^-. \end{cases} \tag{12}$$

We are now ready to describe the algorithm for l^∞ regression.

ALGORITHM 1.
Step 0. Input data $\{(x_i, y_i)\}_1^n$ with $x_1 < x_2 < \cdots < x_n$ and set $I_0^- = 1, I_0^+ = n$, and $i = 0$.
Step 1. While $(I_i^- < I_i^+)$ do
begin

1. compute m_{i+1}, I_{i+1}^- and I_{i+1}^+ with (9)–(12);
2. set i:=i+1;

end { while }.
Step 2. Report the solution $(m^*, b^*) = (m_i, b_i(m_i))$, where $b_i(m_i)$ is computed with (4).

Let us take a look at the complexity issue regarding this algorithm. Define $\text{Gap}(i) \equiv I_i^+ - I_i^-$. Initially, $\text{Gap}(0) = n - 1$. From (11) and (12), we see that $\text{Gap}(i)$ is decreased at least by 1 whenever i is increased by 1. After at most $n-1$ iterations in the while loop, we have $\text{Gap}(n-1) \leq n - 1 - (n-1) = 0$, and therefore, the algorithm is terminated. Thus, the algorithm reports the solution in at most $n - 1$ iterations. At the i^{th} iteration, the major cost is the computation of the m_i which involves at most $2(n - i)$ divisions (c.f., (9),(10)). Thus, the total number of divisions needed by the algorithm to find the solution is no more than $n(n-1)$. Our experience shows that the actual number of iterations in the WHILE loop and the total number of divisions are much less than the theoretical worse case bounds.

4. COMPUTATIONAL RESULTS

First, we illustrate our notation and the algorithm by solving the example given in [3].

EXAMPLE 1. Find the l^∞ regression to the data points

$$(0, 1.520),\ (1, 1.025),\ (2, 0.475),\ (3, 0.010),\ (4, -0.475),\ \text{and}\ (5, -1.005).$$

The solution to this problem is attained after three iterations in the while loop and 17 divisions. We proceed in detail as follows.

Our $\mathcal{F}$ consists of the following members:

$$\begin{aligned}
L_1 &: f_1(m) = 1.520 - 0m,\\
L_2 &: f_2(m) = 1.025 - 1m,\\
L_3 &: f_3(m) = 0.475 - 2m,\\
L_4 &: f_4(m) = 0.010 - 3m,\\
L_5 &: f_5(m) = -0.475 - 4m,\\
L_6 &: f_6(m) = -1.005 - 5m.
\end{aligned}$$

These lines divide the whole plane into the smaller regions as indicated in Figure 1.

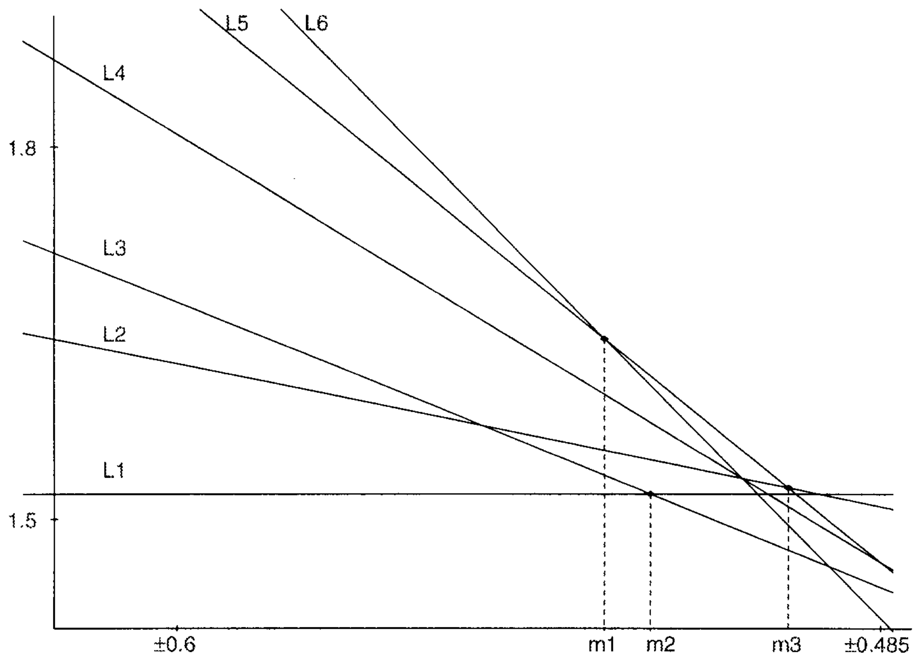

Figure 1. Geometric structure for the ℓ^∞ problem in Example 1.

Initially, $i = 0$, the floor index on interval I_0 is $I_0^- = 1$ and the ceiling index on interval I_0 is $I_0^+ = 6$. Since $I_0^- < I_0^+$, the algorithm proceeds to calculate as in Table 1.

Table 1.

$m_{1,2} = -0.49500$	$m_{6,5} = -0.53000$
$m_{1,3} = -0.52250$	$m_{6,4} = -0.50750$
$m_{1,4} = -0.50333$	$m_{6,3} = -0.49333$
$m_{1,5} = -0.49875$	$m_{6,2} = -0.50750$
$m_{1,6} = -0.50500$	$m_{6,1} = -0.50500$
$m_0^- = m_{1,3} = -0.52250$	$m_0^+ = m_{6,5} = -0.53000$

Thus, $m_1 = \min\{m_0^-, m_0^+\} = -0.53000$, the floor index on interval I_1 remains $I_1^- = 1$, and the ceiling index on interval I_1 changes to $I_1^+ = 5$. Since the floor index did not change, we keep $m_1^- = m_0^- = m_{1,3}$ for the subsequent step.

Next, $i = 1$ and $I_1^- < I_1^+$, so the algorithm continues and calculates as in Table 2.

Thus, $m_2 = \min\{m_1^-, m_1^+\} = -0.52250$, the floor index on interval I_2 is changed to $I_2^- = 3$, and the ceiling index on interval I_2 remains $I_2^+ = 5$.

Table 2.

	$m_{5,4} = -0.48500$
	$m_{5,3} = -0.47500$
	$m_{5,2} = -0.50000$
	$m_{5,1} = -0.49875$
$m_1^- = m_{1,3} = -0.52250$	$m_1^+ = m_{5,2} = -0.50000$

Table 3.

$m_{3,4} = -0.46500$	
$m_{3,5} = -0.47500$	
$m_{3,6} = -0.49333$	
$m_2^- = m_{3,6} = -0.49333$	$m_2^+ = m_{5,2} = -0.50000$

Now $i = 2$ and $I_2^- < I_2^+$, so the algorithm continues and calculates as in Table 3.

Thus, $m_3 = \min\{m_2^-, m_2^+\} = -0.50000$, the floor index on interval I_3 remains $I_3^- = 3$, and the ceiling index on interval I_3 is updated to $I_3^+ = 2$.

We update to $i = 3$, and since $I_3^- > I_3^+$, the algorithm stops. The minimizer (m^*, b^*) is given by

$$m^* = m_3 = -0.5 \qquad \text{and} \qquad b^* = \frac{(y_3 + y_2) - m^*(x_3 + x_2)}{2} = 1.5.$$

Thus, the l^∞ regression line is

$$y = -0.5x + 1.5.$$

We have implemented the algorithm using MATLAB and have applied it to data sets of various random structure in order to study the complexity. One situation of practical interest is as follows. Given a line, form a data set of n equally spaced points on the line, but perturbed by adding a random variable of mean zero to the y-coordinates. For a fixed number of points n, we generated 50 random lines $y = mx$, where $m \sim N(0,1)$ and the corresponding random data sets of the form $\{(x_i, mx_i + \epsilon_i)\}_{i=1}^n$, where $\epsilon_i \sim N(0,1)$. Then we performed the regression using Algorithm 1, and kept track of the number of iterations required by the WHILE loop and the total number of divisions. Table 4 summarizes our results.

Table 4.

n	Average Number of Iterations of the WHILE Loop	Average Total Number of Divisions
100	5.62	980
200	6.34	2268
300	7.28	3948
400	7.38	5416
500	7.38	6759
1000	8.34	15378
2000	8.68	32045
4000	9.62	71912
8000	10.46	157440

It appears that for the randomly generated test problem described above, the average number of iterations of the WHILE loop is around $O(\ln n)$ and the average total number of divisions is around $O(n)$, which are well below the theoretical upper bounds obtained in the previous section. Thus, a probabilistic analysis of the algorithm may be of interest.

5. CONCLUSION AND FINAL REMARKS

Using Lemma 1, it is possible to write a conceptually "straightforward" algorithm to compute the l^∞ regression line. For example, we can compute the abscissa $m_{i,j}$ of all the intersections of all the lines, order these, and then work left to right, determining the floor and ceiling between two successive $m_{i,j}$s (e.g., by evaluating $f_i(\hat{m})$ for all i, where $\hat{m}$ is a test point between two successive $m_{i,j}$s) and then minimize (1) locally using (3). This is computationally much more expensive than Algorithm 1, which takes full advantage of the structure and computes a relatively small number of intersections.

In order to make presentation of Algorithm 1 simple, in this paper, we restricted ourselves to the case $x_1 < x_2 < \cdots < x_n$ and one observation for each distinct point x_i. If the x-coordinates of the data are not in strictly ascending order, then a presort will be necessary. If we allow multiple y-values to correspond to a single x-coordinate, then the solution may not be unique. For example, if our data set is $(0,-1),(0,1),(1,0)$, then any line $y = mx$ with $|m| \leq 1$ will minimize the l^∞ norm of the vector of residuals, hence (1). However, Algorithm 1 can easily be modified to handle this situation and report the entire range of solutions.

Algorithm 1 begins with the leftmost vertex of $z = \psi(m)$, and then inches towards a neighboring vertex on the right, until the optimal vertex is reached. Based on the same technical results of the paper and philosophy of Algorithm 1, we could design an algorithm searching for the optimal vertex in the opposite direction starting from the rightmost vertex of $z = \psi(m)$. It is also possible to design a parallel algorithm searching for the optimizer from both directions if we use a parallel machine.

In addition, it would be interesting to further explore the structure of the problem and try to develp a jump start procedure that would further accelerate the performance of the algorithm presented in this paper.

REFERENCES

1. D.R. Gibson and H. Pulapaka, An algorithm for log rotation in sawmills, *Wood and Fiber Science* **31** (2), 192–199 (1999).
2. I. Barrodale and F.D.K. Roberts, An improved algorithm for discrete l_1 linear approximation, *SIAM J. Numer. Anal.* **10** (5), 839–848 (October 1973).
3. I. Barrodale and A. Young, Algorithms for best L_1 and L_∞ linear approximation on a discrete set, *Numerische Mathematik* **8**, 295–306 (1966).
4. R.H. Bartels, A.R. Conn and C. Charalambous, On the Cline's direct method for solving overdetermined linear systems in the l_∞ sense, *SIAM J. Numer. Anal.* **15**, 255–270 (1978).
5. R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, (1993).